Network Working Group
Request for Comments: 4664
Category: Informational

L. Andersson, Ed.
Acreo AB
E. Rosen, Ed.
Cisco Systems, Inc.

September 2006

Framework for Layer 2 Virtual Private Networks (L2VPNs)

Status of This Memo

Copyright Notice

Abstract

This document provides a framework for Layer 2 Provider Provisioned Virtual Private Networks (L2VPNs). This framework is intended to aid in standardizing protocols and mechanisms to support interoperable L2VPNs.

Table of Contents

1. Introduction

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2. Objectives and Scope of the Document

This document provides a framework for Layer 2 Provider Provisioned Virtual Private Networks (L2VPNs). This framework is intended to aid in standardizing protocols and mechanisms to support interoperable L2VPNs.

The term "provider provisioned VPNs" refers to Virtual Private Networks (VPNs) for which the Service Provider (SP) participates in management and provisioning of the VPN.

Requirements for L2VPNs can be found in [RFC4665].

This document provides reference models for L2VPNs and discusses the functional components of L2VPNs. Specifically, this includes discussion of the technical issues that are important in the design of standards and mechanisms for L2VPNs, including those standards and mechanisms needed for interworking and security.

This document discusses a number of different technical approaches to L2VPNs. It tries to show how the different approaches are related, and to clarify the issues that may lead one to select one approach instead of another. However, this document does not attempt to select any particular approach.

1.3. Layer 2 Virtual Private Networks

There are two fundamentally different kinds of Layer 2 VPN service that a service provider could offer to a customer: Virtual Private Wire Service (VPWS) and Virtual Private LAN Service (VPLS). There is also the possibility of an IP-only LAN-like Service (IPLS).

A VPWS is a VPN service that supplies an L2 point-to-point service. As this is a point-to-point service, there are very few scaling issues with the service as such. Scaling issues might arise from the number of end-points that can be supported on a particular PE.

A VPLS is an L2 service that emulates LAN service across a Wide Area Network (WAN). With regard to the amount of state information that must be kept at the edges in order to support the forwarding function, it has the scaling characteristics of a LAN. Other scaling issues might arise from the number of end-points that can be supported on a particular PE. (See Section 3.4.4.)

Note that VPLS uses a service that does not have native multicast capability to emulate a service that does have native multicast capability. As a result, there will be scalability issues with regard to the handling of multicast traffic in VPLS.

A VPLS service may also impose longer delays and provide less reliable transport than would a native LAN service. The standard LAN control protocols may not have been designed for such an environment and may experience scaling problems when run in that environment.

1.4. Terminology

The list of the technical terms used when discussing L2VPNs may be found in the companion document [RFC4026].

2. Models

2.1. Reference Model for VPWS

The VPWS reference model is shown in Figure 1.

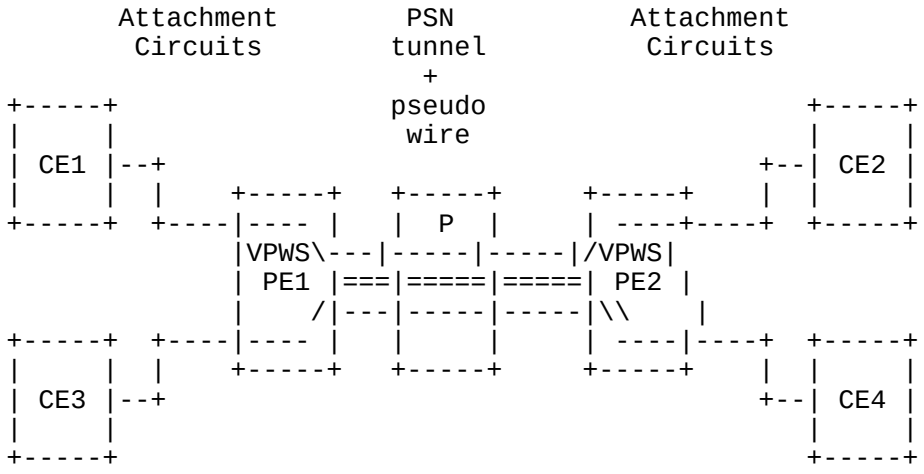


Figure 1

2.1.1. Entities in the VPWS Reference Model

The P, PE (VPWS-PE), and CE devices and the PSN tunnel are defined in [RFC4026]. The attachment circuit and pseudowire are discussed in Section 3. The PE does a simple mapping between the PW and attachment circuit based on local information; i.e., the PW demultiplexor and incoming/outgoing logical/physical port.

2.2. Reference Model for VPLS

The following diagram shows a VPLS reference model where PE devices that are VPLS-capable provide a logical interconnect such that CE devices belonging to a specific VPLS appear to be on a single bridged Ethernet. A VPLS can contain a single VLAN or multiple tagged VLANs.

The VPLS reference model is shown in Figures 2 and 3.

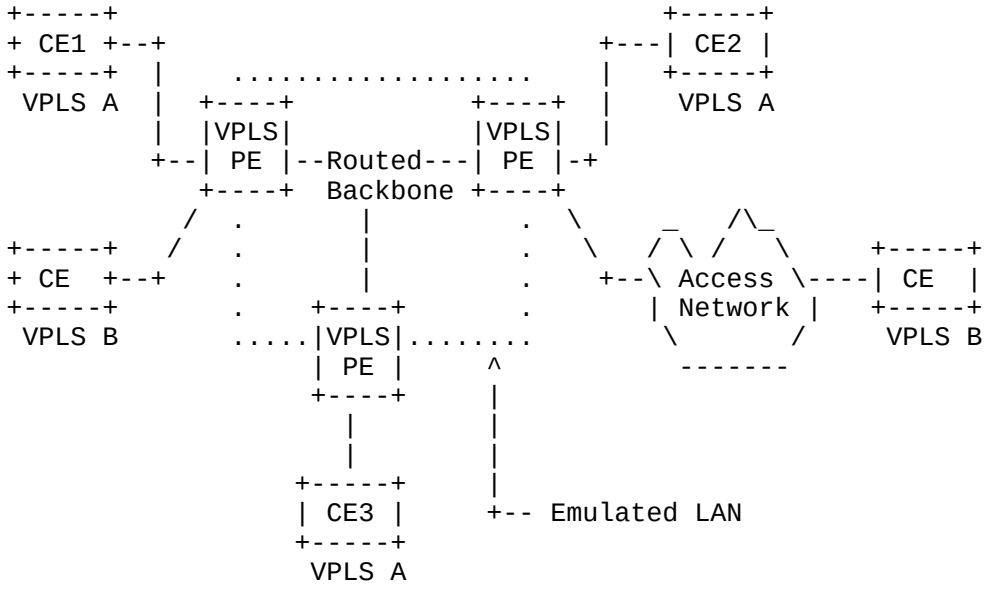


Figure 2

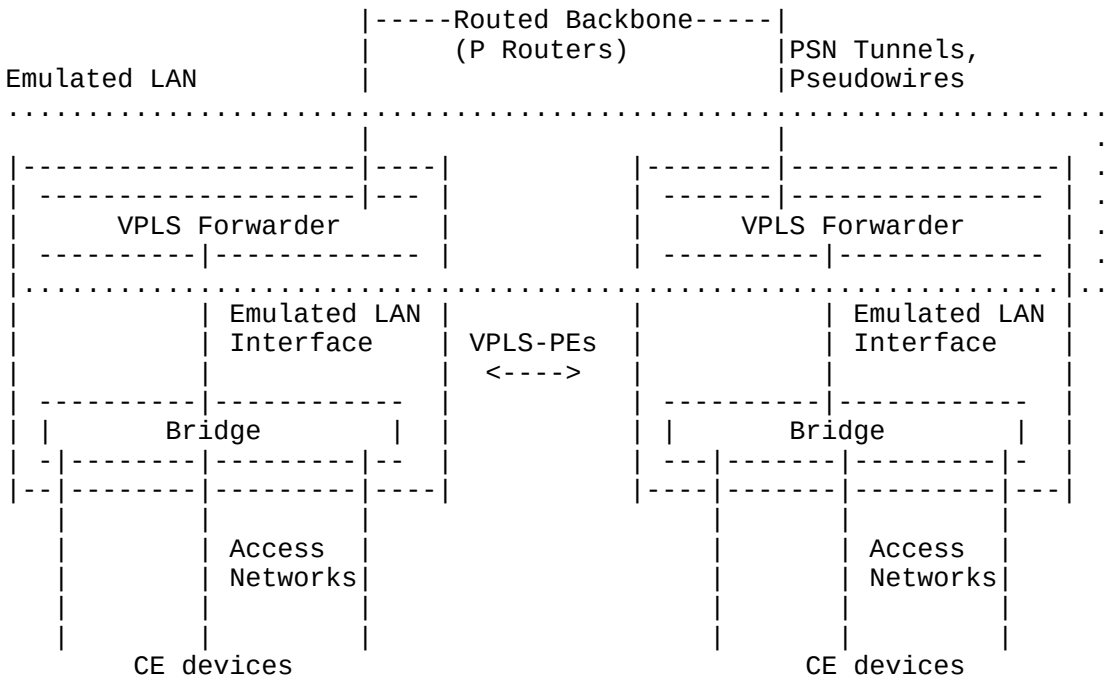


Figure 3

From Figure 3, we see that in VPLS, a CE device attaches, possibly through an access network, to a "bridge" module of a VPLS-PE. Within the VPLS-PE, the bridge module attaches, through an "Emulated LAN Interface", to an Emulated LAN. For each VPLS, there is an Emulated LAN instance. Figure 3 shows some internal structure to the Emulated LAN: it consists of "VPLS Forwarder" modules connected by pseudowires, where the pseudowires may be traveling through PSN tunnels over a routed backbone.

A "VPLS instance" consists of a set of VPLS Forwarders (no more than one per PE) connected by pseudowires.

The functionality that the bridge module must support depends on the service that is being offered by the SP to its customers, as well as on various details of the SP’s network. At a minimum, the bridge module must be able to learn MAC addresses, and to "age them out", in the standard manner. However, if the PE devices have backdoor connections with each other via a Layer 2 network, they may need to be full IEEE bridges ([IEEE8021D]), running a spanning tree with each other. Specification of the precise functionality that the bridge

modules must have in particular circumstances is, however, out of scope of the current document.

This framework specifies that each "bridge module" have a single "Emulated LAN interface". It does not specify the number of bridge modules that a VPLS-PE may contain, nor does it specify the number of VPLS instances that may attach to a bridge module over a single "Emulated LAN interface".

Thus the framework is compatible with at least the following three models:

- Model 1

  A VPLS-PE contains a single bridge module and supports a single VPLS instance. The VPLS instance is an Emulated LAN; if that Emulated LAN contains VLANs, 802.1Q [IEEE8021Q] tagging must be used to indicate which packets are in which VLANs.

- Model 2

  A VPLS-PE contains a single bridge module, but supports multiple VPLS instances. Each VPLS instance is thought of as a VLAN (in effect, an "Emulated VLAN"), and the set of VPLS instances are treated as a set of VLANs on a common LAN. Since each VLAN uses a separate set of PWs, there is no need for 802.1Q tagging.

- Model 3

  A VPLS-PE contains an arbitrary number of bridge modules, each of which attaches to a single VPLS instance.

  There may be other models as well, some of which are combinations of the 3 models above. Different models may have different characteristics, and different scopes of applicability.

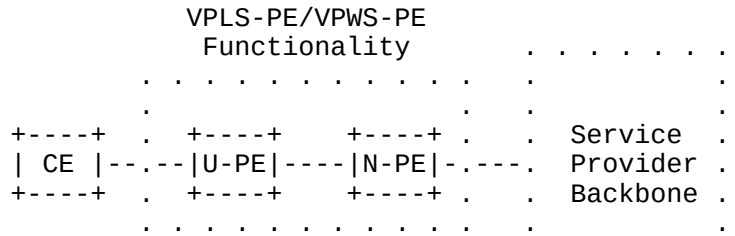
  Each VPLS solution should specify the model or models that it is supporting. Each solution should also specify the necessary bridge functionality that its bridge modules must support.

  This framework does not specify the way in which bridge control protocols are used on the Emulated LANs.

2.2.1. Entities in the VPLS Reference Model

The PE (VPLS-PE) and CE devices are defined in [RFC4026].

2.3. Reference Model for Distributed VPLS-PE or VPWS-PE



2.3.1. Entities in the Distributed PE Reference Models

A VPLS-PE or a VPWS-PE functionality may be distributed to more than one device. The device closer to the customer/user is called the User-facing PE (U-PE), and the device closer to the core network is called Network-facing PE (N-PE).

For further discussion, see Section 3.4.3.

The terms "U-PE" and "N-PE" are defined in [RFC4026].

2.4. VPWS-PE and VPLS-PE

The VPWS-PE and VPLS-PE are functionally very similar, in that they both use forwarders to map attachment circuits to pseudowires. The only difference is that while the forwarder in a VPWS-PE does a one-to-one mapping between the attachment circuit and pseudowire, the forwarder in a VPLS-PE is a Virtual Switching Instance (VSI) that maps multiple attachment circuits to multiple pseudowires (for further discussion, see Section 3).

3. Functional Components of L2 VPN

This section specifies a functional model for L2VPN, which allows one to break an L2VPN architecture down into its functional components. This exhibits the roles played by the various protocols and mechanisms, and thus makes it easier to understand the differences and similarities between various proposed L2VPN architectures.

Section 3.1 contains an overview of some different types of L2VPNs. In Section 3.2, functional components that are common to the different types are discussed. Then, there is a section for each of the L2VPN service types being considered. The latter sections discuss functional components, which may be specific to particular L2VPN types, and type-specific features of the generic components.

3.1. Types of L2VPN

The types of L2VPN are distinguished by the characteristics of the service that they offer to the customers of the Service Provider (SP).

3.1.1. Virtual Private Wire Service (VPWS)

In a VPWS, each CE device is presented with a set of point-to-point virtual circuits.

The other end of each virtual circuit is another CE device. Frames transmitted by a CE on such a virtual circuit are received by the CE device at the other end-point of the virtual circuit. Forwarding from one CE device to another is not affected by the content of the frame, but is fully determined by the virtual circuit on which the frame is transmitted. The PE thus acts as a virtual circuit switch.

This type of L2VPN has long been available over ATM and Frame Relay backbones. Providing this type of L2VPN over MPLS and/or IP backbones is the current topic.

Requirements for this type of L2VPN are specified in [RFC4665].

3.1.2. Virtual Private LAN Service (VPLS)

In a VPLS, each CE device has one or more LAN interfaces that lead to a "virtual backbone".

Two CEs are connected to the same virtual backbone if and only if they are members of the same VPLS instance (i.e., same VPN). When a CE transmits a frame, the PE that receives it examines the MAC Destination Address field in order to determine how to forward the frame. Thus, the PE functions as a bridge. As Figure 3 indicates, if a set of PEs support a common VPLS instance, then there is an Emulated LAN, corresponding to that VPLS instance, to which each of those PE bridges attaches (via an emulated interface). From the perspective of a CE device, the virtual backbone is the set of PE bridges and the Emulated LAN on which they reside. Thus to a CE device, the LAN that attaches it to the PE is extended transparently over the routed MPLS and/or IP backbone.

The PE bridge function treats the Emulated LAN as it would any other LAN to which it has an interface. Forwarding decisions are made in the manner that is normal for bridges, which is based on MAC Source Address learning.

VPLS is like VPWS in that forwarding is done without any consideration of the Layer3 header. VPLS is unlike VPWS in that:

- VPLS allows the PE to use addressing information in a frame's L2 header to determine how to forward the frame; and

- VPLS allows a single CE/PE connection to be used for transmitting frames to multiple remote CEs; in this particular respect, VPLS resembles L3VPN more than VPWS.

Requirements for this type of L2VPN are specified in [RFC4665].

3.1.3. IP-Only LAN-Like Service (IPLS)

An IPLS is very like a VPLS, except that:

- it is assumed that the CE devices are hosts or routers, not switches; and

- it is assumed that the service will only carry IP packets and supporting packets such as ICMP and ARP (in the case of IPv4) or Neighbor Discovery (in the case of IPv6); Layer 2 packets that do not contain IP are not supported.

While this service is a functional subset of the VPLS service, it is considered separately because it may be possible to provide it using different mechanisms, which may allow it to run on certain hardware platforms that cannot support the full VPLS functionality.

3.2. Generic L2VPN Transport Functional Components

All L2VPN types must transport "frames" across the core network connecting the PEs. In all L2VPN types, a PE (PE1) receives a frame from a CE (CE1), and then transports the frame to a PE (PE2), which then transports the frame to a CE (CE2). In this section, we discuss the functional components that are necessary to transport L2 frames in any type of L2VPN service.

3.2.1. Attachment Circuits

In any type of L2VPN, a CE device attaches to a PE device via some sort of circuit or virtual circuit. We will call this an "Attachment Circuit" (AC). We use this term very generally; an Attachment Circuit may be a Frame Relay DLCI, an ATM VPI/VCI, an Ethernet port, a VLAN, a PPP connection on a physical interface, a PPP session from

an L2TP tunnel, an MPLS LSP, etc. The CE device may be a router, a switch, a host, or just about anything, which the customer needs hooked up to the VPN. An AC carries a frame between CE and PE, or vice versa.

Procedures for setting up and maintaining the ACs are out of scope of this architecture.

These procedures are generally specified as part of the specification of the particular Attachment Circuit technology.

Any given frame will traverse an AC from a CE to a PE, and then on another AC from a PE to a CE.

We refer to the former AC as the frame's "ingress AC" and to the latter AC as the frame's "egress AC". Note that this notion of "ingress AC" and "egress AC" is relative to a specific frame and denotes nothing more than the frame's direction of travel while it is on that AC.

3.2.2. Pseudowires

A "Pseudowire" (PW) is a relation between two PE devices. Whereas an AC is used to carry a frame from CE to PE, a PW is used to carry a frame between two PEs. We use the term "pseudowire" in the sense of [RFC3985].

Setting up and maintaining the PWs is the job of the PEs. State information for a particular PW is maintained at the two PEs that are its endpoints, but not at other PEs, and not in the backbone routers (P routers).

Pseudowires may be point-to-point, multipoint-to-point, or point-to-multipoint. In this framework, point-to-point PWs are always considered bidirectional; multipoint-to-point and point-to-multipoint PWs are always considered unidirectional. Multipoint-to-point PWs can be used only when the PE receiving a frame does not need to infer, from the PW on which the frame was received, the identity of the frame's ingress AC. Point-to-multipoint PWs may be useful when frames need to be multicast.

Procedures for setting up and maintaining point-to-multipoint PWs are not considered in this version of this framework.

Any given frame travels first on its ingress AC, then on a PW, and then on its egress AC.

Multicast frames may be replicated by a PE, so of course the information carried in multicast frames may travel on more than one PW and more than one egress AC.

Thus with respect to a given frame, a PW may be said to associate a number of ACs. If these ACs are of the same technology (e.g., both ATM, both Ethernet, both Frame Relay), the PW is said to provide "homogeneous transport"; otherwise it is said to provide "heterogeneous transport". Heterogeneous transport requires that some sort of interworking function be applied. There are at least three different approaches to interworking:

1. One of the CEs may perform the interworking locally. For example, if CE1 attaches to PE1 via ATM, but CE2 attaches to PE2 via Ethernet, then CE1 may decide to send/receive Ethernet frames over ATM, using the RFC 2684, "LLC Encapsulation for Bridged Protocols". In such a case, PE1 would need to know that it is to terminate the ATM VC locally, and only to send/receive Ethernet frames over the PW.

2. One of the PEs may perform the interworking. For example, if CE1 attaches to PE1 via ATM, but CE2 attaches to PE2 via Frame Relay, PE1 may provide the "ATM/FR Service Interworking" function. This would be transparent to the CEs, and the PW would carry only Frame Relay frames.

3. IPLS could be used. In this case, the "frames" carried by the PW are IP datagrams, and the two PEs need to cooperate in order to spoof various L2-specific procedures used by IP (see Section 3.5).

If heterogeneous PWs are used, the setup protocol must ensure that each endpoint knows the MTU of the remote AC. If the two ACs do not have the same MTU, one of the following three procedures must be carried out:

- The PW is not allowed to come up.

- The endpoint at the AC with the larger MTU must reduce the AC’s MTU so that it is the same as the MTU of the remote AC.

- The two endpoints must agree to use a specified fragmentation/reassembly procedure.

3.2.3. Forwarders

In all types of L2VPN, a PE (say, PE1) receives a frame over an AC and forwards it over a PW to another PE (say, PE2). PE2 then forwards the frame out on another AC.

The case in which PE1 and PE2 are the same device is an important case to handle correctly, in order to provide the L2VPN service properly. However, as this case does not require any protocol, we do not address it further in this document.

When PE1 receives a frame on a particular AC, it must determine the PW on which the frame must be forwarded. In general, this is done by considering:

- the incoming AC;
- possibly the contents of the frame's Layer2 header; and
- possibly some forwarding information that may be statically or dynamically maintained.

If dynamic or static forwarding information is considered, the information is specific to a particular L2VPN instance (i.e., to a particular VPN).

Similarly, when PE2 receives a frame on a particular PW, it must determine the AC on which the frame must be forwarded. This is done by considering:

- the incoming PW;
- possibly the contents of the frame's Layer2 header; and
- possibly some forwarding information that may be statically or dynamically maintained.

If dynamic or static forwarding information is considered, the information is specific to a particular L2VPN instance (i.e., to a particular VPN).

The procedures used to make the forwarding decision are known as a "forwarder". We may think of a PW as being "bound", at each of its endpoints, to a forwarder. The forwarder in turn "binds" the PWs to ACs. Different types of L2VPN have different types of forwarders.

For instance, a forwarder may bind a single AC to a single PW, ignoring all frame contents and using no other forwarding information. Or a forwarder may bind an AC to a set of PWs and ACs, moving individual frames from AC to PW, from a PW to an AC or from AC to AC by comparing information from the frame's Layer2 header to information in a forwarding database. This is discussed in more detail below, as we consider the different L2VPN types.

3.2.4. Tunnels

A PW is carried in a "tunnel" from PE1 to PE2. We assume that an arbitrary number of PWs may be carried in a single tunnel; the only requirement is that the PWs all terminate at PE2.

We do not even require that all the PWs in the tunnel originate at PE1; the tunnels may be multipoint-to-point tunnels. Nor do we require that all PWs between the same pair of PEs travel in the same tunnel. All we require is that when a frame traveling through such a tunnel arrives at PE2, PE2 will be able to associate it with a particular PW.

(While one can imagine tunneling techniques that only allow one PW per tunnel, they have evident scalability problems, and we do not consider them further.)

A variety of different tunneling technologies may be used for the PE-PE tunnels. All that is really required is that the tunneling technologies allow the proper demultiplexing of the contained PWs. The tunnels might be MPLS LSPs, L2TP tunnels, IPsec tunnels, MPLS-in-IP tunnels, etc. Generally the tunneling technology will require the use of an encapsulation that contains a demultiplexor field, where the demultiplexor field is used to identify a particular PW. Procedures for setting up and maintaining the tunnels are not within the scope of this framework. (But see Section 3.2.6, "Pseudowire Signaling".)

If there are multiple tunnels from PE1 to PE2, it may be desirable to assign a particular PE1-PE2 PW to a particular tunnel based on some particular characteristics of the PW and/or the tunnel. For example, perhaps different tunnels are associated with different QoS characteristics, and different PWs require different QoS. Procedures for specifying how to assign PWs to tunnels are out of scope of the current framework.

Though point-to-point PWs are bidirectional, the tunnels in which they travel need not be either bidirectional or point-to-point. For example, a point-to-point PW may travel within a unidirectional multipoint-to-point MPLS LSP.

3.2.5. Encapsulation

As L2VPN packets are carried in pseudowires, standard pseudowire encapsulation formats and techniques (as specified by the IETF's PWE3 WG) should be used wherever applicable.

Generally the PW encapsulations will themselves be encapsulated within a tunnel encapsulation, as determined by the specification of the tunneling protocol.

It may be necessary to define additional PW encapsulations to cover areas that are of importance for L2VPN, but that may not be within the scope of PWE3. Heterogeneous transport may be an instance of this.

3.2.6. Pseudowire Signaling

Procedures for setting up and maintaining the PWs themselves are within the scope of this framework. This includes procedures for distributing demultiplexor field values, even though the demultiplexor field, strictly speaking, belongs to the tunneling protocol and not to the PW.

The signaling for a point-to-point pseudowire must perform the following functions:

- Distribution of the demultiplexor.

  Since many PWs may be carried in a single tunnel, the tunneling protocol must assign a demultiplexor value to each PW. These demultiplexors must be unique with respect to a given tunnel (or, with some tunneling technologies, unique at the egress PE). Generally, the PE that is the egress of the tunnel will select the demultiplexor values and will distribute them to the PE(s) which is (are) the ingress(es) of the tunnel. This is the essential part of the PW setup procedure.

  Note that, as is usually the case in tunneling architectures, the demultiplexor field belongs to the tunneling protocol, not to the protocol being tunneled. For this reason, the PW setup protocols may be extensions of the control protocols for setting up the tunnels.

- Selection of the Forwarder at the remote PE.

  The signaling protocol must contain enough information to enable the remote PE to select the proper forwarder to which the PW is to be bound. We can call this information the "Remote Forwarder

Selector". The information that is required will depend on the type of L2VPN being provided and on the provisioning model being used (see Sections 3.3.1 and 3.4.2). The Remote Forwarder Selector may uniquely identify a particular Forwarder, or it may identify an attribute of Forwarders. In the latter case, it would select whichever Forwarder has been provisioned with that attribute.

- Supporting pseudowire emulations.

  To the extent that a particular PW must emulate the signaling of a particular Layer2 technology, the PW signaling must provide the necessary functions.

- Distribution of state changes.

  Changes in the state of an AC may need to be reflected in changes to the state of the PW to which the AC is bound, and vice versa. The specification as to which changes need to be reflected in what way would generally be within the province of the PWE3 WG.

- Establishing pseudowire characteristics.

  To the extent that one or more characteristics of a PW must be known to and/or agreed upon by both endpoints, the signaling must allow for the necessary interaction.

As specified above, signaling for point-to-point PWs must pass enough information to allow a remote PE to properly bind a PW to a Forwarder, and to associate a particular demultiplexor value with that PW. Once the two PEs have done the proper PW/Forwarder bindings, and have agreed on the demultiplexor values, the PW may be considered set up. If it is necessary to negotiate further characteristics or parameters of a particular PW, or to pass status information for a particular PW, the PW may be identified by the demultiplexor value.

Signaling procedures for point-to-point pseudowires are most commonly point-to-point procedures that are executed by the two PW endpoints. There are, however, proposals to use point-to-multipoint signaling for setting up point-to-point pseudowires, so this is included in the framework. When PWs are themselves point-to-multipoint, it is also possible to use either point-to-point signaling or point-to-multipoint signaling to set them up. This is discussed in the remainder of this section.

3.2.6.1. Point-to-Point Signaling

There are several ways to do the necessary point-to-point signaling. Among them are:

- LDP

  LDP [RFC3036] extensions can be defined for pseudowire signaling. This form of signaling can be used for pseudowires that are to be carried in MPLS "tunnels", or in MPLS-in-something-else tunnels.

- L2TP

  L2TP [RFC2661] can be used for pseudowire signaling, resulting in pseudowires that are carried as "sessions" within L2TP tunnels. Pseudowire-specific extensions to L2TP may also be needed.

Other methods may be possible as well.

It is possible to have one control connection between a pair of PEs, which is used to control many PWs.

The use of point-to-point signaling for setting up point-to-point PWs is straightforward. Multipoint-to-point PWs can also be set up by point-to-point signaling, as the remote PEs do not necessarily need to know whether the PWs are multipoint-to-point or point-to-point. In some signaling procedures, the same demultiplexor value may be assigned to all the remote PEs.

3.2.6.2. Point-to-Multipoint Signaling

Consider the following conditions:

- It is necessary to set up a set of PWs, all of which have the same characteristics.

- It is not necessary to use the PW signaling protocol to pass PW state changes.

- For each PW in the set, the same value of the Remote Forwarder Selector can be used.

Call these the "Environmental Conditions".

Suppose also that there is some mechanism by which, given a range of demultiplexor values, each of a set of PEs can make a unique and

deterministic selection of a single value from within that range. Call this the "Demultiplexor Condition". Alternatively, suppose that one is trying to set up a multipoint-to-point PW rather than to set up a point-to-point PW. Call this the "Multipoint Condition".

If:

- The Environmental Conditions hold; and
- Either
  * the Demultiplexor Condition holds, or
  * the Multipoint Condition holds,

then for a given set of PWs that terminate at egress PE1, the information that PE1 needs to send to the ingress PE(s) of each pseudowire in the set is exactly the same. All the ingress PE(s) receive the same Forwarder Selector value. They all receive the same set of PW parameters (if any). And either they all receive the same demultiplexor value (if the PW is multipoint-to-point) or they all receive a range of demultiplexor values from which each can choose a unique demultiplexor value for itself.

Rather than connect to each ingress PE and replicate the same information, it may make sense either to multicast the information, or to send the information once to a "reflector", which will then take responsibility for distributing the information to the other PEs.

We refer to this sort of technique as "point-to-multipoint" signaling. It would, for example, be possible to use BGP [RFC1771] to do the signaling, with PEs that are BGP peers not of each other, but of one or more BGP route reflectors [RFC2796].

3.2.6.3. Inter-AS Considerations

Pseudowires may need to run from a PE in one Service Provider's network to a PE in another Service Provider's network. This has the following implications:

- The signaling protocol that sets up the PWs must be able to cross network boundaries. Of course, all IP-based protocols have this capability.
- The two PEs at the PW endpoints must be addressable and routable from each other.

- The signaling protocol needs to allow each PW endpoint to authenticate the other. To make use of the authentication capability, there would also need to be some method of key distribution that is acceptable to both administrations.

3.2.7. Service Quality

Service Quality refers to the ability for the network to deliver a Service level Specification (SLS) for service attributes such as protection, security, and Quality of Service (QoS). The service quality provided depends on the subscriber's requirements and can be characterized by a number of performance metrics.

The necessary Service Quality must be provided on the ACs, as well as on the PWs. Mechanisms for providing Service Quality on the PWs may be PW-specific or tunnel-specific; in the latter case, the assignment of a PW to a tunnel may depend on the Service Quality.

3.2.7.1. Quality of Service (QoS)

QoS describes the queuing behavior applied to a particular "flow", in order to achieve particular goals of precedence, throughput, delay, jitter, etc.

Based on the customer Service Level Agreement (SLA), traffic from a customer can be prioritized, policed, and shaped for QoS requirements. The queuing and forwarding policies can preserve the packet order and QoS parameters of customer traffic. The class of services can be mapped from information in the customer frames, or it can be independent of the frame content.

QoS functions can be listed as follows:

- Customer Traffic Prioritization: L2VPN services could be best effort or QoS guaranteed. Traffic from one customer might need to be prioritized over others when sharing same network resources. This requires capabilities within the L2VPN solution to classify and mark priority to QoS guaranteed customer traffic.

- Proper queuing behavior would be needed at the egress AC, and possibly within the backbone network as well. If queuing behavior must be controlled within the backbone network, the control might be based on CoS information in the MPLS or IP header, or it might be achieved by nesting particular tunnels within particular traffic engineering tunnels.

- Policing: This ensures that a user of L2VPN services uses network resources within the limits of the agreed SLA. Any excess L2VPN traffic can be rejected or handled differently based on provider policy.

- Policing would generally be applied at the ingress AC.

- Shaping: Under some cases, the random nature of L2VPN traffic might lead to sub-optimal utilization of network resources. Through queuing and forwarding mechanisms, the traffic can be shaped without altering the packet order.

- Shaping would generally be applied at the ingress AC.

3.2.7.2. Resiliency

Resiliency describes the ability of the L2VPN infrastructure to protect a flow from network outage, so that service remains available in the presence of failures.

L2VPN, like any other service, is subject to failures such as link, trunk, and node failures, both in the SP's core network infrastructure and on the ACs.

It is desirable that the failure be detected "immediately" and that protection mechanisms allow fast restoration times to make L2VPN service almost transparent to these failures to the extent possible, based on the level of resiliency. Restoration should take place before the CEs can react to the failure. Essential aspects of providing resiliency are:

- Link/Node failure detection: Mechanisms within the L2VPN service should allow for link or node failures that impact the service, and that should be detected immediately.

- Resiliency policy: The way in which a detected failure is handled will depend on the restoration policy of the SLA associated with the L2VPN service specification. It may need to be handled immediately, or it may need to be handled only if no other critical failure needs protection resources, or it may be completely ignored if it is within the bounds of the "acceptable downtime" allowed by the L2VPN service.

- Restoration Mechanisms: The L2VPN solutions could allow for physical level protection, logical level protection, or both. For example, by connecting customers over redundant and

physically separate ACs to different provider customer-facing devices, one AC can be maintained as active, and the other could be marked as a backup; upon the failure detection across the primary AC, the backup could become active.

To a great extent, resiliency is a matter of having appropriate failure and recovery mechanisms in the network core, including "ordinary" adaptive routing as well as "fast reroute" capabilities. The ability to support redundant ACs between CEs and PEs also plays a role.

3.2.8. Management

An L2VPN solution can provide mechanisms to manage and monitor different L2VPN components. From a Service Level Agreement (SLA) perspective, L2VPN solutions could allow monitoring of L2VPN service characteristics and offer mechanisms used by Service Providers to report such monitored statistical data. Trouble-shooting and verification of operational and maintenance activities of L2VPN services are essential requirements for Service Providers.

3.3. VPWS

A VPWS is an L2VPN service in which each forwarder binds exactly one AC to exactly one PW. Frames received on the AC are transmitted on the PW; frames received on the PW are transmitted on the AC. The content of a frame's Layer2 header plays no role in the forwarding decision, except insofar as the Layer2 header contents are used to associate the frame with a particular AC (e.g., the DLCI field of a Frame Relay frame identifies the AC).

A particular combination of <AC, PW, AC> forms a "virtual circuit" between two CE devices.

A particular VPN (VPWS instance) may be thought of as a collection of such virtual circuits, or as an "overlay" of PWs on the MPLS or IP backbone. This creates an overlay topology that is in effect the "virtual backbone" of a particular VPN.

Whether two virtual circuits are said to belong to the same VPN or not is an administrative matter based on the agreements between the SPs and their customers. This may impact the provisioning model (discussed below). It may also affect how particular PWs are assigned to tunnels, the way QoS is assigned to particular ACs and PWs, etc.

Note that VPWS makes use of point-to-point PWs exclusively.

3.3.1. Provisioning and Auto-Discovery

Provisioning a VPWS is a matter of:

1. Provisioning the ACs;

2. Providing the PEs with the necessary information to enable them to set up PWs between ACs to result in the desired overlay topology; and

3. Configuring the PWs with any necessary characteristics.

3.3.1.1. Attachment Circuit Provisioning

In many cases, the ACs must be individually provisioned on the PE and/or CE. This will certainly be the case if the CE/PE attachment technology is a switched network, such as ATM or FR, and the VCs are PVCs rather than SVCs. It is also the case whenever the individual Attachment Circuits need to be given specific parameters (e.g., QoS parameters, guaranteed bandwidth parameters) that differ from circuit to circuit.

There are also cases in which ACs might not have to be individually provisioned. For example, if an AC is just an MPLS LSP running between a CE and a PE, it could be set up as the RESULT of setting up a PW rather than having to be provisioned BEFORE the PW can be set up. The same may apply whenever the AC is a Switched Virtual Circuit of any sort, though in this case, various policy controls might need to be provisioned; e.g., limiting the number of ACs that can be set up between a given CE and a given PE.

Issues such as whether the Attachment Circuits need to be individually provisioned or not, whether they are Switched VCs or Permanent VCs, and what sorts of policy controls may be applied are implementation and deployment issues and are considered to be out of scope of this framework.

3.3.1.2. PW Provisioning for Arbitrary Overlay Topologies

In order to support arbitrary overlay topologies, it is necessary to allow the provisioning of individual PWs. In this model, when a PW is provisioned on a PE device, it is locally bound to a specific AC. It is also provisioned with information that identifies a specific AC at a remote PE.

There are basically two variations of this provisioning model:

- Two-sided provisioning

  With two-sided provisioning, each PE that is at the end of a PW is provisioned with the following information:

  * Identifier of the Local AC to which the PW is to be bound

  * PW type and parameters

  * IP address of the remote PE (i.e., the PE that is to be at the remote end of the PW)

  * Identifier that is meaningful to the remote PE, and that can be passed in the PW signaling protocol to enable the remote PE to bind the PW to the proper AC. This can be an identifier of the PW or an identifier of the remote AC. If a PW identifier is used, it must be unique at each of the two PEs. If an AC identifier is used, it need only be unique at the remote PE.

  This identifier is then used as the Remote Forwarder Selector when signaling is done (see 3.2.6.1).

- Single-sided provisioning

  With single-sided provisioning, a PE at one end of a PW is provisioned with the following information:

  * Identifier of the Local AC to which the PW is to be bound

  * PW type and parameters

  * Globally unique identifier of remote AC

  This identifier is then used as the Forwarder Selector when signaling is done (see section 3.2.6.1).

  In this provisioning model, the IP address of the remote PE is not provisioned. Rather, the assumption is that an auto-discovery scheme will be used to map the globally unique identifier to the IP address of the remote PE, along with an identifier (perhaps unique only at the latter PE) for an AC at that PE. The PW signaling protocol can then make a connection to the remote PE, passing the AC identifier, so that the remote PE binds the PW to the proper AC.

This scheme requires provisioning of the PW at only one PE, but it does not eliminate the need (if there is a need) to provision the ACs at both PEs.

These provisioning models fit well with the use of point-to-point signaling. When each PW is individually provisioned, as the conditions necessary for the use of point-to-multipoint signaling do not hold.

3.3.1.3. Colored Pools PW Provisioning Model

Suppose that at each PE, sets of ACs are gathered together into "pools", and that each such pool is assigned a "color". (For example, a pool might contain all and only the ACs from this PE to a particular CE.) Now suppose that we impose the following rule: whenever PE1 and PE2 have a pool of the same color, there will be a PW between PE1 and PE2 that is bound at PE1 to an arbitrarily chosen AC from that pool, and at PE2 to an arbitrarily chosen AC from that pool. (We do not rule out the case where a single PE has multiple pools of a given color.)

For example, each pool in a particular PE might represent a particular CE device, for which the ACs in the pool are the ACs connecting that CE to that PE. The color might be a VPN-id. Application of this provisioning model would then lead to a full CE-to-CE mesh within the VPN, where every CE in the VPN has a virtual circuit to every other CE within the VPN.

More specifically, to provision VPWS according to this model, one provisions a set of pools and configures each pool with the following information:

- The set of ACs that belong to the pool (with no AC belonging to more than one pool)

- The color

- A pool identifier that is unique at least relative to the color.

  An auto-discovery procedure is then used to map each color into a list of ordered pairs <IP address of PE, pool id>. The occurrence of a pair <X, Y> on this list means that the PE at IP address X has a pool with pool id Y, which is of the specified color.

This information can be used to support several different signaling techniques. One possible technique proceeds as follows:

- A PE finds that it has a pool of color C.
- Using auto-discovery, it obtains the set of ordered pairs <X,Y> for color C.
- For each such pair <X,Y>, it:
  * removes an AC from the pool;
  * binds the AC to a particular PW; and
  * signals PE X via point-to-point signaling that the PW is to be bound to an AC from pool Y.

Another possible signaling technique is the following:

- A PE finds that it has a pool of color C, containing n ACs.
- It binds each AC to a PW, creating a set of PWs. This set of PWs is then organized into a sequence. (For instance, each PW may be associated with a demultiplexor field value, and the PWs may then be sequenced according to the numerical value of their respective demultiplexors.)
- Using auto-discovery, it obtains the list of PE routers that have one or more pools of color C.
- It signals each such PE router, specifying the sequence Q of PWs.
- If PE X receives such a signal and PE X has a pool Y of the specified color, it:
  * removes an AC from the pool; and
  * binds the AC to the PW that is the "Yth" PW in the sequence Q.

This presumes, of course, that the pool identifiers are or can be uniquely mapped into small ordinal numbers; assigning the pool identifiers in this way becomes a requirement of the provisioning system.

Note that since this technique signals the same information to all the remote PEs, it can be supported via point-to-multipoint signaling.

This provisioning model can be applied as long as the following conditions hold:

- There is no need to provision different characteristics for the different PWs;

- It makes no difference which pairs of ACs are bound together by PWs, as long as both ACs in the pair come from like-colored pools; and

- It is possible to construct the desired overlay topology simply by assigning colors to the pools. (This is certainly simple if a full mesh is desired, or if a hub and spoke configuration is desired; creating arbitrary topologies is less simple, and is perhaps not always possible.)

3.3.2. Requirements on Auto-Discovery Procedures

Some of the requirements for auto-discovery procedures can be deduced from the above.

To support the single-sided provisioning model, auto-discovery must be able to map a globally unique identifier (of a PW or of an Attachment Circuit) to an IP address of a PE.

To support the colored pools provisioning model, auto-discovery must enable a PE to determine the set of other PEs that contain pools of the same color.

These requirements enable the auto-discovery scheme to provide the information, which the PEs need to set up the PWs.

There are additional requirements on the auto-discovery procedures that cannot simply be deduced from the provisioning model:

- Particular signaling schemes may require additional information before they can proceed and hence may impose additional requirements on the auto-discovery procedures.

- A given Service Provider may support several different types of signaling procedures, and thus the PEs may need to learn, via auto-discovery, which signaling procedures to use.

- Changes in the configuration of a PE should be reflected by the auto-discovery procedures, within a timely manner, and without the need to explicitly reconfigure any other PE.

- The auto-configuration procedures must work across service provider boundaries. This rules out, e.g., use of schemes that piggyback the auto-discovery information on the backbone's IGP.

3.3.3. Heterogeneous Pseudowires

Under certain circumstances, it may be desirable to have a PW that binds two ACs that use different technologies (e.g., one is ATM, one is Ethernet). There are a number of different ways, depending on the AC types, in which this can be done. For example:

- If one AC is ATM and one is FR, then standard ATM/FR Network Interworking can be used. In this case, the PW might be signaled for ATM, where the Interworking function occurs between the PW and the FR AC.

- A common encapsulation can be used on both ACs, if for example, one AC is Ethernet and one is FR, an "Ethernet over FR" encapsulation can be used on the latter. In this case, the PW could be signaled for Ethernet, with processing of the Ethernet over FR encapsulation local to the PE with the FR AC.

- If it is known that the two ACs attach to IP routers or hosts and carry only IP traffic, then one could use a PW that carries the IP packets, and the respective Layer2 encapsulations would be local matters for the two PEs. However, if one of the ACs is a LAN and one is a point-to-point link, care would have to be taken to ensure that procedures such as ARP and Inverse ARP are properly handled; this might require some signaling, and some proxy functions. Further, if the CEs use a routing algorithm that has different procedures for LAN interfaces than those for point-to-point interfaces, additional mechanisms may be required to ensure proper interworking.

3.4. VPLS Emulated LANs

A VPLS is an L2VPN service in which:

- the ACs attach CE devices to PE bridge modules; and
- each PE bridge module is attached via an "emulated LAN interface" to an "emulated LAN".

This is shown in Figure 3.

In this section, we examine the functional decomposition of the VPLS Emulated LAN. An Emulated LAN's ACs are the "emulated LAN interfaces" attaching PE bridge modules to the "VPLS Forwarder" modules (see Figure 3). The payload on the ACs consists of ethernet frames, with or without VLAN headers.

A given VPLS Forwarder in a given PE will have multiple ACs only if there are multiple bridge modules in that PE that attach to that Forwarder. This scenario is included in the Framework, though discussion of its utility is out of scope.

The set of VPLS Forwarders within a single VPLS are connected via PWs. Two VPLS Forwarders will have a PW between them only if those two Forwarders are part of the same VPLS. (There may be a further restriction that two VPLS Forwarders have a PW between them only if those two Forwarders belong to the same VLAN in the same VPN.) A particular set of interconnected VPLS Forwarders is what constitutes a VPLS Emulated LAN.

On a real LAN, any frame transmitted by one entity is received by all the others. A VPLS Emulated LAN, however, behaves somewhat differently. When a VPLS Forwarder receives a unicast frame over one of its Emulated LAN interfaces, the Forwarder does not necessarily send the frame to all the other Forwarders on that Emulated LAN. A unicast frame needs to be sent to only one other Forwarder in order to be properly delivered to its destination MAC address. If the transmitting Forwarder knows which other Forwarder needs to receive a particular unicast frame, it will send the frame to just that one Forwarder. This forwarding optimization is an important part of any attempt to provide a VPLS service over a wide-area or metropolitan area network.

In effect, then, each Forwarder behaves as a "Virtual Switch Instance" (VSI), maintaining a forwarding table that maps MAC addresses to PWs. The VSI is populated in much the same way that a standard bridge populates its forwarding table. The VPLS Forwarders do MAC Source Address (SA) learning on frames received on PWs from

other Forwarders and must also do the related set of procedures, such as aging out address entries. Frames with unknown DAs or multicast DAs must be "broadcast" by one Forwarder to all the others (on the same emulated LAN). There are, however, a few important differences between the VPLS Forwarder VSI and the standard bridge forwarding function:

- A VPLS Forwarder never learns the MAC SAs of frames that it receives on its ACs; it only learns the MAC SAs of frames that are received on PWs from other VPLS Forwarders; and

- The VPLS Forwarders of a particular emulated LAN do not participate in a spanning tree protocol with each other. A "split horizon" technique is used to prevent forwarding loops.

These points are discussed further in the next section.

Note that the PE bridge modules that are on a given Emulated LAN may or may not run a spanning tree protocol with each other over the Emulated LAN; whether they do so or not is outside the scope of the VPLS specifications. The PE bridge modules will do MAC address learning on the ACs. The PE bridge modules also do MAC address learning on the Emulated LAN interfaces, but do not do MAC address learning on the PWs, as the PWs are "hidden" behind the Emulated LAN interface. Conceptually, the PE bridge module's forwarding table and the VPLS Forwarder's VSI are distinct entities. (Of course, particular implementations might combine these into a single table, but that is beyond the scope of this document.)

A further issue arises if the PE bridges run bridge control protocols with each other over the Emulated LAN. Bridge control protocols are generally designed to run in over a real LAN and may presume, for their proper functioning, certain characteristics of the LAN, such as low latency and sequential delivery. If the Emulated LAN does not provide these characteristics, the control protocols may not perform as expected unless special mechanisms are provided for carrying the control frames.

It should be noted that changes in the spanning tree (if any) of a customer network, or in the spanning tree (if any) of the PE bridges, may cause certain MAC addresses to change their location from one PE to another. These changes may not be visible to the VPLS Forwarders, which means that those MAC addresses might become unreachable until they are aged out of the first PE's VSI. If this is not acceptable, some mechanism for communicating such changes to the VPLS Forwarders must be provided.

3.4.1. VPLS Overlay Topologies and Forwarding

Within a single VPLS, the VPLS Forwarders are interconnected by PWs. The set of PWs thus forms an "overlay topology".

The VPLS Forwarder VSIs are populated by means of MAC address learning. That is, the VSI keeps track of which MAC SAs have been received over which PWs. The presumption, of course, is that if a particular MAC address appears as the SA of a frame received over a particular PW, then frames that carry that MAC address in the DA field should be sent to the VSI that is at the remote end of the PW. In order for this presumption to be true, there must be a unique VSI at the remote end of the PW, which means that VSIs cannot be interconnected by means of multipoint-to-point PWs. The PWs are necessarily either point-to-point or, possibly, point-to-multipoint.

MAC learning over a point-to-point PW is done via the standard techniques as specified by IEEE, where the PW is treated by the VPLS Forwarder as a "bridge port". Of course, if a MAC address is learned from a point-to-multipoint PW, the VSI must indicate that packets to that address are to be sent over a point-to-point PW that leads to the root of that point-to-multipoint PW.

The VSI forwarding decisions must be coordinated so that loop-free forwarding over the overlay topology is ensured.

There are several possible types of overlay topologies:

- Full mesh

  In a full mesh, every VSI in a given VPLS has exactly one point-to-point PW to every other VSI in that same VPLS.

  In this topology, loop free forwarding of frames is ensured by the following rule: if a VSI receives a frame, over a PW, from another VSI, it MUST NOT forward that frame over ANY other PW to any other VSI. This ensures that once a frame traverses the Emulated LAN, it must be sent off the Emulated LAN.

  If a VSI receives, on one of its Emulated LAN interfaces, a unicast frame with a known DA, the frame is sent on exactly one point-to-point PW.

  If a VSI receives, on one of its Emulated LAN interfaces, a multicast frame or a unicast frame with an unknown DA, it sends a copy of the frame to each other VSI in the same Emulated LAN. This can be done by replicating the frame and sending a copy over each point-to-point PW. Alternatively, the full mesh of

point-to-point PWs may be augmented with point-to-multipoint PWs, where each VSI in a VPLS is the transmitter on a single point-to-multipoint PW, and the receivers on that PW are all the other VSIs in that VPLS.

- Tree structured

  In a tree structured topology, every VSI in a particular VPLS is provisioned to be at a particular level in the tree. A given VSI has at most one pseudowire leading to a higher level. The root of the tree is considered the highest level.

  In this topology, loop free forwarding of frames is ensured by the following rule: if a frame is received over a pseudowire from a higher level, it may not be sent over a pseudowire that leads to a higher level.

- Tree with Meshed Highest Level

  In this variant of the tree-structured topology, there may be more than one VSI at the highest level, but the set of VSIs that are at the highest level must be fully meshed. To ensure loop free forwarding, we need to impose the rule that a frame can be sent on a pseudowire to the same or higher level only if it arrived over a pseudowire from a lower level, and that frames arriving over PWs from the same level cannot be sent on PWs to the same level.

Other overlay topologies are also possible; e.g., an arbitrary partial mesh of PWs among the VSIs of a VPLS. Loop-freedom could then be assured by, for example, running a spanning tree on the overlay. These topologies are not further considered in this framework.

Note that loop freedom in the overlay topology does not necessarily ensure loop freedom in the overall customer LAN that contains the VPLS. It does not even ensure loop freedom among the PE bridge modules. It ensures only that when a frame is sent on the Emulated LAN, the frame will not loop endlessly before (or instead of) leaving the Emulated LAN.

Improper configuration of the customer LAN or PE bridge modules may cause frames to loop, and frames that fall into such loops may transit the overlay topology multiple times. Procedures that enable the PE to detect and/or prevent such loops may be advisable.

3.4.2. Provisioning and Auto-Discovery

Each VPLS must be assigned a globally unique identifier. This can be thought of as a VPN-id.

The ACs attaching the CEs to the PEs must be provisioned on both the PEs and the CEs. A VSI for that VPLS must be provisioned on the PE, and the local ACs of that VPLS must be associated with that VSI. The VSI must be provisioned with the identifier of the VPLS to which it belongs.

An auto-discovery scheme may be used by a PE to map a VPLS identifier into the set of remote PEs that have VSIs in that VPLS. Once this set is determined, the PE can use pseudowire signaling to set up a PW to each of those VSIs. The VPLS identifier would serve as the signaling protocol's Forwarder Selector. This would result in a full mesh of PWs among the VSIs in a particular VPLS.

If a single VPLS contains multiple VLANs, then it may be desirable to limit connectivity so that two VSIs are connected only if they have a VLAN in common.

In this case, each VSI would need to be provisioned with one or more VLAN ids, and the auto-discovery scheme would need to map a VPLS identifier into pairs of <PE, VLAN id>.

If a fully meshed topology of VSIs is not desired, then each VSI needs to be provisioned with additional information specifying its placement in the topology. This information would also need to be provided by the auto-discovery scheme.

Alternatively, the single-sided provisioning method discussed in Section 3.3.1.2 could be used. As this is more complicated, it would only be used if it were necessary to associate individual PWs with individual characteristics. For example, if different guaranteed bandwidths were needed between different pairs of sites within a VPLS, the PWs would have to be provisioned individually.

3.4.3. Distributed PE

Often, when a VPLS type of service is provided, the CE devices attach to a provider-managed CPE device. This provider-managed CPE device may attach to CEs of multiple customers, especially if, for example, there are multiple customers occupying the same building. However, this device is really part of the SP's network, hence may be considered a PE device.

In some scenarios in which a VPLS type of service is provided, the CE devices attach to a provider-managed intermediary device. This provider-managed device may attach to CEs of multiple customers. This may arise if there are multiple customers occupying the same building. This device is really part of the SP's network and may for that reason be considered to be a PE device; however, in the simplest case, it is performing only aggregation and none of the function associated with a VPLS.

Relative to the VPLS there are three different possibilities for allocate functions to a device in such a position in the provider network:

- it can perform aggregation and pure Layer2 service only, in which case it does not really play the role of a PE device in a VPLS service. In this case the intermediary system must connect to devices that perform VPLS PE functionality; the intermediary device itself is not part of the VPLS architecture and has hence not been named in this architecture.

- it can perform all the PE functions relevant for a VPLS. In such a case, the device is called VPLS-PE, see [RFC4026]. This type of device will be connected to the core (P) routers.

The PE functionality for a VPLS may be distributed between two devices, one "low-end" closer to the customer that performs, for example, the MAC-address learning and forwarding decisions, and one "high-end" that performs the control functions; e.g., establishing tunnels, PWs, and VCs. We call the low-end device the User-Facing PE (U-PE) and the high-end device the Network-Facing PE (N-PE).

It is conceivable that the U-PE may be placed very close to the customer; e.g., in a building with more than one customer. The N-PE will presumably be placed on the SP's premises.

The distributed case is potentially of interest for a number of possible reasons:

- The N-PE may be a device that cannot easily implement the VSI functionality described above. For example, perhaps the N-PE is a router that cannot perform the high speed MAC learning that is needed in order to implement a VSI forwarder. At the same time, the U-PE may need to be a low-cost device that also cannot implement the full set of VPLS functions.

This leads one to investigate further if there are sensible ways to split the VPLS PE functionality between the U-PE and the N-PE.

- Generally, in the L2VPN architecture, the PEs are expected to participate as peers in the backbone routing protocol. Since the number of U-PEs is potentially very large relative to the number of N-PEs, this may be undesirable as a matter of scaling the backbone routing protocol.

- The U-PE may be a relatively inexpensive device that is unable to participate in the full range of signaling and/or auto-discovery procedures that are needed in order to provide the VPLS service.

The VPLS functionality can be distributed between U-PE and N-PE in a number of different ways, and a number of different proposals have been made. They all presume that the U-PE will maintain a VSI forwarder, connected by PWs to the remote VSIs; the N-PE thus does not need to perform the VSI forwarding function. The proposals tend to differ with respect to the following questions:

- Should the U-PEs perform full PW signaling to set up the PWs to remote VSIs, or should the N-PEs do this signaling?

  Since the U-PEs need to be able to send packets on PWs to remote VSIs and receive packets on PWs from remote VSIs, if the PW signaling is done by the N-PE, there would have to be some form of "lightweight" (presumably) signaling between N-PE and U-PE that allows the PWs to be extended from N-PE to U-PE.

- Should the U-PEs do their own auto-discovery, or should this be done by the N-PEs?

  In the latter case, the U-PEs may need to have some means of telling the N-PEs which VPLSes they are interested in, and the N-PEs must have some means of passing the results of the auto-discovery process to the U-PE.

  Whether it makes sense to split auto-discovery in this manner may depend on the particular auto-discovery protocol used. One would not expect the U-PEs to participate in, if for example, a BGP-based auto-discovery scheme, but perhaps they would be expected to participate in a RADIUS-based auto-discovery scheme.

- If a U-PE does not participate in routing but is redundantly connected to two different N-PEs, can the U-PE still make an intelligent choice of the best N-PE to use as the "next hop" for

traffic destined to a particular remote VSI? If not, can this choice be made as the result of some other sort of interaction between N-PE and U-PE, or does this choice need to be established by provisioning?

- If a U-PE does not participate in routing but does participate in full PW signaling, and if MPLS is being used, how can an N-PE send a U-PE the labels that the U-PE needs in order to be able to send traffic to its signaling peers? (If the U-PE did participate in routing, this would happen automatically.)

- When a frame must be multicast, should the replication be done by the N-PE or the U-PE?

These questions are not all independent; the way one answers some of them may influence the way one answers others.

3.4.4. Scaling Issues in VPLS Deployment

In general, the PSN supports a VPLS solution with a tunnel from each VPLS-PE to every other VPLS-PE participating in the same VPLS instance. Strictly, VPLS-PEs with more than one VPLS instance in common only need one tunnel, but for resource allocation reasons it might be necessary to establish several tunnels. For each VPLS service on a given VPLS-PE, it needs to establish one pseudowire to every other VPLS-PE participating in that VPLS service. In total n*(n-1) pseudowires must be setup between the VPLS-PE routers. In large scale deployment this obviously creates scaling problems. One way to address the scaling problems is to use hierarchy.

3.5. IP-Only LAN-Like Service (IPLS)

If, instead of providing a general VPLS service, one wishes to provide a VPLS that is used only to connect IP routers or hosts (i.e., the CE devices are all assumed to be IP routers or hosts), then it is possible to make certain simplifications.

In this environment, all Ethernet frames sent from a particular CE to a particular PE on a particular Attachment Circuit will have the same MAC Source Address. Thus, rather than use address learning in the data plane to learn the MAC addresses, the PE can use the control plane to learn the MAC address. This allows the PE to be implemented on devices that are not capable of doing MAC address learning in the data plane.

To eliminate the need for MAC address learning on the PWs as well as on the ACs, the pseudowire signaling protocol would have to carry the MAC address from one pseudowire endpoint to the other. In the case

of IPv4, Each PE would perform proxy ARP to its directly attached CEs. In the case of IPv6, each PE would send proxy Neighbor and/or Router Advertisements.

Eliminating the need to do MAC address learning on the PWs eliminates the need for the PWs to be point-to-point. Multipoint-to-point PWs could be used instead.

Unlike a VPLS, all the ACs in an IPLS would not necessarily have to carry Ethernet frames; only the IP packets would need to be passed across the network, not their Layer 2 wrappers. However, if there are protocols that are specific to the Layer 2, but that provide, for example, address resolution services for Layer 3, it may then be necessary to "translate" (or otherwise interwork) one of these Layer 2 protocols to the other. For example, if an IPLS instance has an ethernet AC and a Frame Relay AC, and IPv4 is running on both, interworking between ARP and Inverse ARP might be required.

The set of routing protocols that could be carried across the IPLS might also be restricted.

An IPLS instance must have a particular IPLS-wide MTU; if there are different kinds of AC in an IPLS instance, and those different kinds of AC support different MTUs, all ACS must enforce the IPLS-wide MTU; an AC that cannot do this must not be allowed to join the IPLS instance.

4. Security Considerations

The security considerations section of the L2VPN requirements document [RFC4665] addresses a number of areas that are potentially insecure aspects of the L2VPN. These relate to both control plane and data plane security issues that may arise in the following areas:

- issues fully contained in the provider network
- issues fully contained in the customer network
- issues in the customer-provider interface network

These three areas are addressed below.

4.1. Provider Network Security Issues

This section discusses security issues that only impact the SP’s equipment.

There are security issues having to do with the control connections that are used on a PE-PE basis for setting up and maintaining the pseudowires.

A PE should not engage with another PE in a control connection unless it has some confidence that the peer is really a PE to which it should be setting up PWs. Otherwise, L2PVN traffic may go to the wrong place. If control packets are maliciously and undetectably altered while in flight, denial of service, or alteration of the expected quality of service, may result.

If peers discover each other dynamically (via some auto-discovery procedure), this presupposes that the auto-discovery procedures are themselves adequately trusted.

PEs should not accept control connections from arbitrary entities; a PE either should be configured with its peers or should learn them from a trusted auto-configuration procedure. If the peer is required to be within the same SP's network, then access control filters at the borders of that network can be used to prevent spoofing of the peer's source address. If the peer is from another SP's network, then setting up such filters may be difficult or even impossible, depending on the way in which the two SPs are connected. Even if the access filters can be set up, the level of assurance that they provide will be lower.

Thus, for inter-SP control connections, it is advisable to use some sort of cryptographic authentication procedure. Control protocols which used TCP may use the TCP MD5 option to provide a measure of PE-PE authentication; this requires at least one shared secret between SPs. The use of IPsec between PEs is also possible and provides a greater degree of assurance, though at a greater cost.

Any other security considerations that apply to the control protocol in general will also apply when the control protocol is used for setting up PWs. If the control protocol uses UDP messages, it may be advisable to have some protection against spoofed UDP messages that appear to be from a valid peer; this requires further study.

To limit the effect of Denial of Service attacks on a PE, some means of limiting the rate of processing of control plane traffic may be desirable.

Unlike authentication and integrity, privacy of the signaling messages is not usually considered very important. If it is needed, the signaling messages can be sent through an IPsec connection.

If the PE cannot efficiently handle high volumes of multicast traffic for sustained periods, then it may be possible to launch a denial of service attack on a VPLS service by sending a PE a large number of frames that have either a multicast address or an unknown MAC address in their MAC Destination Address fields. A similar denial of service attack can be mounted by sending a PE a large number of frames with bogus MAC Source Address fields. The bogus addresses can fill the MAC address tables in the PEs, with the result that frames destined to the real MAC addresses always get flooded (i.e., multicast). Note that this flooding can remove the (weak) confidentiality property of this or any other bridged network.

4.2. Provider-Customer Network Security Issues

There are a number of security issues related to the access network between the provider and the customer. This is also traditionally a network that is hard to protect physically.

Typical security issues on the provider-customer interface include the following:

- Ensuring that the correct customer interface is configured
- Preventing unauthorized access to the PE
- Preventing unauthorized access to a specific PE port
- Ensuring correct service delimiting fields (VLAN, DLCI, etc.)

As the access network for an L2VPN service is necessarily a Layer 2 network, it is preferable to use authentication mechanisms that do not presuppose any IP capabilities on the CE device.

There are existing Layer 2 protocols and best current practices to guard against these security issues. For example, IEEE 802.1x defines authentication at the link level for access through an ethernet bridge; the Frame Relay Forum defines LMI extensions for authentication (FRF.17).

4.3. Customer Network Security Issues

Even if all CE devices are properly authorized to attach to their PE devices, misconfiguration of the PE may interconnect CEs that are not supposed to be in the same L2VPN.

In a VPWS, the CEs may run IPsec to authenticate each other. Other Layer 3 or Layer 4 protocols may have their own authentication methods.

In a VPLS, CE-to-CE IPsec is even more problematic, as IPsec does not well support the multipoint configuration that is provided by the VPLS service.

There may be alternative methods for achieving a degree of CE-to-CE authentication, if the L2VPN signaling protocol can carry opaque objects between the CEs, either inband (over the L2VPN) or out-of-band, through the participation of the signaling protocol. This is for further study.

The L2VPN procedures do not provide authentication, integrity, or privacy for the customer's traffic; if this is needed, it becomes the responsibility of the customer. For customers who really need these features or who do not trust their service providers to provide the level of security that they need, the L2VPN framework discussed in this document may not be satisfactory. Such customers may consider alternative L2VPN schemes that are based not on an overlay of PWs, but on an overlay of IPsec tunnels whose endpoints are at the customer sites; however, such alternatives are not discussed in this document.

If there is CE-to-CE control traffic (e.g., BPDUs) on whose integrity the customer's own Layer 2 network depends, it may be advisable to send the control traffic using some more secure mechanism than is used for the data traffic.

In general, any means of mounting a denial of service attack on bridged networks generally can also be used to mount a denial of service attack on the VPLS service for a particular customer. We have discussed here only those attacks that rely on features of the VPLS service that are not shared by bridged networks in general.

5. Acknowledgements

This document is the outcome of discussions within a Layer 2 VPN design team, all of whose members could be considered co-authors. Specifically, the co-authors are Loa Andersson, Waldemar Augustyn, Marty Borden, Hamid Ould-Brahim, Juha Heinanen, Kireeti Kompella, Vach Kompella, Marc Lasserre, Pascal Menezes, Vasile Radoaca, Eric Rosen, and Tissa Senevirathne.

The authors would like to thank Marco Carugi for cooperation in setting up context, working directions, and taking time for discussions in this space; Tove Madsen and Pekka Savola for valuable input and reviews; and Norm Finn, Matt Squires, and Ali Sajassi for valuable discussion of the VPLS issues.

6. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3985] Bryant, S. and P. Pate, "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, March 2005.

[RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, March 2005.

[RFC4665] Augustyn, W., Ed. and Y. Serbest, Ed., "Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks (L2VPNs)", RFC 4665, September 2006.

7. Informative References

[IEEE8021D] IEEE 802.1D-2003, "IEEE Standard for Local and Metropolitan Area Networks: Media Access Control (MAC) Bridges"

[IEEE8021Q] IEEE 802.1Q-1998, "IEEE Standards for Local and Metropolitan Area Networks: Virtual Bridged Local Area Networks"

[RFC1771] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[RFC2661] Townsley, W., Valencia, A., Rubens, A., Pall, G., Zorn, G., and B. Palter, "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[RFC2796] Bates, T., Chandra, R., and E. Chen, "BGP Route Reflection - An Alternative to Full Mesh IBGP", RFC 2796, April 2000.

[RFC3036] Andersson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", RFC 3036, January 2001.

Authors' Addresses

Loa Andersson
Acreo AB

EMail: loa@pi.se

Eric C. Rosen
Cisco Systems, Inc.
1414 Massachusetts Avenue
Boxborough, MA 01719

EMail: erosen@cisco.com

Waldemar Augustyn

EMail: waldemar@wdmsys.com

Marty Borden

EMail: mborden@acm.org

Juha Heinanen
Song Networks, Inc.
Hallituskatu 16
33200 Tampere, Finland

EMail: jh@song.fi

Kireeti Kompella
Juniper Networks, Inc.
1194 N. Mathilda Ave
Sunnyvale, CA 94089

EMail: kireeti@juniper.net

Vach Kompella
TiMetra Networks
274 Ferguson Dr.
Mountain View, CA 94043

EMail: vach.kompella@alcatel.com

Marc Lasserre
Riverstone Networks
5200 Great America Pkwy
Santa Clara, CA 95054

EMail: mlasserre@lucent.com

Pascal Menezies

EMail: pascalm1@yahoo.com

Hamid Ould-Brahim
Nortel Networks
P O Box 3511 Station C
Ottawa, ON K1Y 4H7, Canada

EMail: hbrahim@nortelnetworks.com

Vasile Radoaca
Nortel Networks
600 Technology Park
Billerica, MA 01821

EMail: radoaca@hotmail.com

Tissa Senevirathne
1567 Belleville Way
Sunnyvale CA 94087

EMail: tsenevir@hotmail.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).